

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号
特開2002-24858
(P2002-24858A)

(43)公開日 平成14年 1 月25日 (2002. 1. 25)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 T 17/40		G 0 6 T 17/40	A 2 C 0 0 1
A 6 3 F 13/00		A 6 3 F 13/00	C 5 B 0 5 0
G 0 6 T 15/00	1 0 0	G 0 6 T 15/00	1 0 0 A 5 B 0 8 0

審査請求 未請求 請求項の数14 O L (全 25 頁)

(21)出願番号 特願2000-205355(P2000-205355)

(22)出願日 平成12年 7 月 6 日 (2000. 7. 6)

(71)出願人 000134855

株式会社ナムコ

東京都大田区多摩川 2 丁目 8 番 5 号

(72)発明者 指田 容子

東京都大田区多摩川 2 丁目 8 番 5 号 株式
会社ナムコ内

(74)代理人 100090387

弁理士 布施 行夫 (外 2 名)

F ターム(参考) 2C001 BA00 BA06 BC00 BC10 CA01

CA05 CA07 CB01 CC02 CC08

5B050 BA08 CA07 EA21 EA28

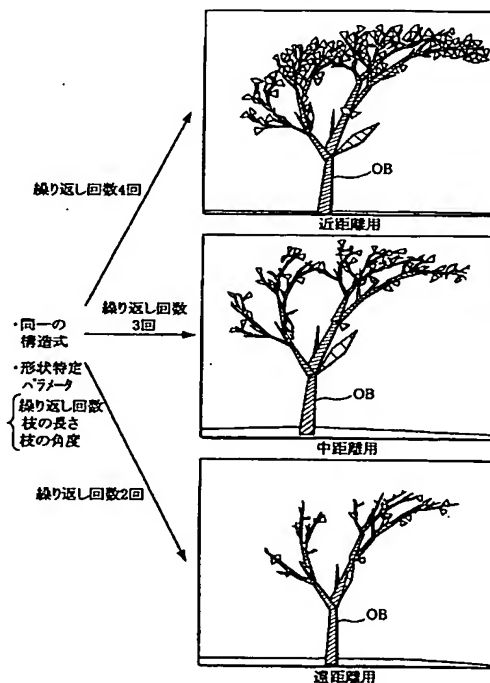
5B080 AA13 CA07 FA14 GA18 GA22

(54)【発明の名称】 ゲームシステム及び情報記憶媒体

(57)【要約】

【課題】 自己相似形の再帰的生成を利用した画像生成処理を、少ない処理負担と少ない使用記憶容量で実現できるゲームシステム、情報記憶媒体の提供。

【解決手段】 Lシステム等を利用して自己相似形を所与の繰り返し回数だけ再帰的に生成する。自己相似形の生成の繰り返し回数を視点からの距離に応じて変化させる。視点からの距離に応じて変化する繰り返し回数と同一の再帰型構造式に基づいて、視点からの距離に応じて精密度が変化するオブジェクトの画像を生成する。再帰型構造式や形状特定パラメータをゲーム状況の変化に応じてリアルタイムに変化させる。ゲーム状況の変化に応じて、再帰型構造式のコードの書き換え、新たなコードの付加、コードの削除を行う。処理対象となっているコードが再帰呼び出しのコードであった場合には、再帰呼び出しを実行して呼び出し先の構造式のコードの処理に移行し、全てのコードの処理が終了した後に、呼び出し元の構造式のコードの処理に戻る。



【特許請求の範囲】

【請求項1】 画像生成を行うゲームシステムであって、
自己相似形を所与の繰り返し回数だけ再帰的に生成する手段と、
自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、
前記繰り返し回数を視点からの距離に応じて変化させる手段と、
を含むことを特徴とするゲームシステム。

【請求項2】 請求項1において、
再帰型の構造式と、前記繰り返し回数を含む形状特定パラメータとに基づいて自己相似形を再帰的に生成する場合において、
視点からの距離に応じて変化する前記繰り返し回数と、
同一の前記再帰型構造式とに基づいて、視点からの距離に応じて精密度が変化するオブジェクトの画像を生成することを特徴とするゲームシステム。

【請求項3】 画像生成を行うゲームシステムであって、
再帰型の構造式と形状特定パラメータとに基づいて自己相似形を再帰的に生成する手段と、
自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、
前記再帰型構造式又は前記形状特定パラメータをゲーム状況の変化に応じてリアルタイムに変化させる手段と、
を含むことを特徴とするゲームシステム。

【請求項4】 請求項3において、
ゲーム状況の変化に応じて、前記再帰型構造式を含むコードを書き換える、又は前記再帰型構造式に新たなコードを付加する、又は前記再帰型構造式を含むコードを削除することを特徴とするゲームシステム。

【請求項5】 請求項3又は4において、
前記形状特定パラメータが、自己相似形の再帰的な生成の繰り返し回数、自己相似形を構成するエッジの長さ、又はエッジの分岐角度を含む場合において、
前記繰り返し回数、前記エッジの長さ、又は前記エッジの分岐角度をゲーム状況の変化に応じて変化させることを特徴とするゲームシステム。

【請求項6】 請求項3乃至5のいずれかにおいて、
自己相似形を再帰的に生成することで得られる情報に基づいて、オブジェクトを構成する基本パーツを配置する場合において、
前記基本パーツをゲーム状況の変化に応じて差し替えることを特徴とするゲームシステム。

【請求項7】 画像生成を行うゲームシステムであって、
再帰型の構造式と形状特定パラメータとに基づいて自己

相似形を再帰的に生成する手段と、
自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段とを含み、
処理対象となっている再帰型構造式のコードが再帰呼び出しを必要とするコードであった場合には、再帰呼び出しを実行して呼び出し先の再帰型構造式のコードの処理に移行し、呼び出し先の再帰型構造式の全てのコードの処理が終了した後に、呼び出し元の再帰型構造式のコードの処理に戻ることを特徴とするゲームシステム。

【請求項8】 コンピュータが使用可能な情報記憶媒体であって、
自己相似形を所与の繰り返し回数だけ再帰的に生成する手段と、
自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、
前記繰り返し回数を視点からの距離に応じて変化させる手段と、
を実行するためのプログラムを含むことを特徴とする情報記憶媒体。

【請求項9】 請求項8において、
再帰型の構造式と、前記繰り返し回数を含む形状特定パラメータとに基づいて自己相似形を再帰的に生成する場合において、
視点からの距離に応じて変化する前記繰り返し回数と、
同一の前記再帰型構造式とに基づいて、視点からの距離に応じて精密度が変化するオブジェクトの画像を生成することを特徴とする情報記憶媒体。

【請求項10】 コンピュータが使用可能な情報記憶媒体であって、
再帰型の構造式と形状特定パラメータとに基づいて自己相似形を再帰的に生成する手段と、
自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、
前記再帰型構造式又は前記形状特定パラメータをゲーム状況の変化に応じてリアルタイムに変化させる手段と、
を実行するためのプログラムを含むことを特徴とする情報記憶媒体。

【請求項11】 請求項10において、
ゲーム状況の変化に応じて、前記再帰型構造式を含むコードを書き換える、又は前記再帰型構造式に新たなコードを付加する、又は前記再帰型構造式を含むコードを削除することを特徴とする情報記憶媒体。

【請求項12】 請求項10又は11において、
前記形状特定パラメータが、自己相似形の再帰的な生成の繰り返し回数、自己相似形を構成するエッジの長さ、又はエッジの分岐角度を含む場合において、
前記繰り返し回数、前記エッジの長さ、又は前記エッジ

の分岐角度をゲーム状況の変化に応じて変化させることを特徴とする情報記憶媒体。

【請求項13】 請求項10乃至12のいずれかにおいて、

自己相似形を再帰的に生成することで得られる情報に基づいて、オブジェクトを構成する基本パーツを配置する場合において、

前記基本パーツをゲーム状況の変化に応じて差し替えることを特徴とする情報記憶媒体。

【請求項14】 コンピュータが使用可能な情報記憶媒体であって、

再帰型の構造式と形状特定パラメータとに基づいて自己相似形を再帰的に生成する手段と、

自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、

を実行するためのプログラムを含み、

処理対象となっている再帰型構造式のコードが再帰呼び出しを必要とするコードであった場合には、再帰呼び出しを実行して呼び出し先の再帰型構造式のコードの処理に移行し、呼び出し先の再帰型構造式の全てのコードの処理が終了した後に、呼び出し元の再帰型構造式のコードの処理に戻ることを特徴とする情報記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ゲームシステム及び情報記憶媒体に関する。

【0002】

【背景技術及び発明が解決しようとする課題】従来より、仮想的な3次元空間であるオブジェクト空間内の所与の視点から見える画像を生成するゲームシステムが知られており、いわゆる仮想現実を体験できるものとして人気が高い。レーシングゲームを楽しむことができるゲームシステムを例にとれば、プレーヤは、レーシングカー（モデル）を操作してオブジェクト空間内で走行させ、他のプレーヤやコンピュータが操作するレーシングカーと競争することで3次元ゲームを楽しむ。

【0003】このようなゲームシステムでは、プレーヤの仮想現実感の向上のために、よりリアルな画像を生成することが重要な技術的課題になっている。従って、マップに配置される樹木などの植物についても、よりリアルな画像で表現できることが望まれる。

【0004】そして、このような植物の枝や葉の付き方をリアルにシミュレートできる手法として、植物学者A. Lindenmayerが考案したLシステムと呼ばれる手法が従来より知られている。このLシステムでは、植物の生成規則を記述した構造式と形状特定パラメータ（枝の長さ、枝の分岐角度等）とに基づき自己相似形を再帰的に生成することで、植物の枝や葉の付き方をリアルにシミュレートする。

【0005】しかしながら、このLシステムは、リアルタイム処理が要求されないCG（Computer Graphics）の分野では用いられていたが、リアルタイム処理が要求されるゲームシステムにおいては未だ用いられていなかった。そして、Lシステムの手法をゲームシステムにおいて実現するためには、如何にして少ない処理負担と少ない使用記憶容量で、Lシステムを利用した植物の画像生成を実現できるかが重要な技術的課題になる。

【0006】ところで、ゲームシステムにおける画像生成処理の負担を軽減化する手法として、視点（仮想カメラ）からの距離に応じてモデルを切り替えるLOD（Level of Detail）と呼ばれる手法が従来より知られている。

この従来のLODでは、図1（A）に示すように、視点とオブジェクトとの距離が近い場合にはポリゴン数が多いモデルでオブジェクトを表し、視点とオブジェクトとの距離が遠い場合にはポリゴン数が少ないモデルでオブジェクトを表す。これにより、得られる画像の品質を維持しながら、1フレーム内に描画すべきポリゴンの数を減らすことができる。

【0007】しかしながら、この従来のLODでは、オブジェクトを表すモデルとして例えば近距離用、中距離用、遠距離用モデルというように複数のモデルを用意しなければならない。従って図1（B）に示すように、モデルデータの記憶に必要な使用記憶容量が増大化してしまうという問題点がある。

【0008】また、従来のLODでは、モデル切り替えの際の画面のショックが大きいう問題点もある。例えば、中距離用モデルから近距離用モデルに切り替える際に、モデルが切り替わったことをプレーヤに気づかれてしまうと、プレーヤの仮想現実感を大幅に損ねてしまう。そして、この問題を解決する手法として、第1の距離用、第2の距離用、第3の距離用・・・第Mの距離用モデルというように、用意するモデルの数を増やす手法も考えられる。しかしながら、この手法によると、モデルデータの使用記憶容量が増大化するという上記問題点は、更に深刻なものになる。

【0009】本発明は、以上のような課題に鑑みてなされたものであり、その目的とするところは、自己相似形の再帰的生成を利用した画像生成処理を、少ない処理負担と少ない使用記憶容量で実現できるゲームシステム及び情報記憶媒体を提供することにある。

【0010】

【課題を解決するための手段】上記課題を解決するため、本発明は、画像生成を行うゲームシステムであって、自己相似形を所与の繰り返し回数だけ再帰的に生成する手段と、自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、前記繰り返し回数を視点からの距離に応じて変化させる手段とを含むことを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使

用可能な情報記憶媒体であって、上記手段を実行するためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、上記手段を実行するための処理ルーチンを含むことを特徴とする。

【0011】本発明によれば、自己相似形が所与の繰り返し回数だけ再帰的に生成される。そして、この生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像が生成される。例えば、自己相似形の再帰的生成により得られた情報に基づいて基本パーツを配置することで画像が生成されたり、上記情報に基づいてテクスチャを生成し、生成されたテクスチャを利用して画像が生成される。

【0012】そして本発明では、上記繰り返し回数が視点からの距離に応じて変化する。これにより、オブジェクトの画像の精密度の変化等を、繰り返し回数を変化させるだけという少ない処理負担で実現できるようになる。

【0013】なお視点からの距離は、奥行き距離でもよいし、視点とオブジェクトとの直線距離でもよいし、これらの距離と均等なパラメータでもよい。

【0014】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、再帰型の構造式と、前記繰り返し回数を含む形状特定パラメータとに基づいて自己相似形を再帰的に生成する場合において、視点からの距離に応じて変化する前記繰り返し回数と、同一の前記再帰型構造式とに基づいて、視点からの距離に応じて精密度が変化するオブジェクトの画像を生成することを特徴とする。

【0015】このようにすれば、同一の再帰型構造式を用いて、オブジェクトの画像の精密度を変化させることが可能になる。従って、少ない使用記憶容量で、よりリアルな画像を生成できるようになる。

【0016】また本発明は、画像生成を行うゲームシステムであって、再帰型の構造式と形状特定パラメータとに基づいて自己相似形を再帰的に生成する手段と、自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段と、前記再帰型構造式又は前記形状特定パラメータをゲーム状況の変化に応じてリアルタイムに変化させる手段とを含むことを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であって、上記手段を実行するためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、上記手段を実行するための処理ルーチンを含むことを特徴とする。

【0017】本発明によれば、再帰型構造式と形状特定パラメータとに基づいて自己相似形が再帰的に生成され

る。そして、この生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像が生成される。

【0018】そして本発明では、上記再帰型構造式又は上記形状特定パラメータがゲーム状況の変化に応じてリアルタイムに変化する。従って、ゲーム状況に応じた適切な画像を、再帰型構造式や形状特定パラメータを変化させるだけという簡素な処理で生成できるようになり、少ない処理負担でリアルな画像を生成できる。

【0019】なおゲーム状況が変化すると、例えば、ゲーム世界において種々のイベントが発生する場合や、ゲーム処理における種々の判断に使用されるパラメータが変化する場合などである。

【0020】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、ゲーム状況の変化に応じて、前記再帰型構造式を含むコードを書き換える、又は前記再帰型構造式に新たなコードを付加する、又は前記再帰型構造式を含むコードを削除することを特徴とする。

【0021】このようにすれば、コードを書き換え、付加、削除などの簡素な処理で、季節が変化する様子や、樹木が生長する様子などを表現できるようになり、少ない処理負担でよりリアルな画像を生成できる。

【0022】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、前記形状特定パラメータが、自己相似形の再帰的な生成の繰り返し回数、自己相似形を構成するエッジの長さ、又はエッジの分岐角度を含む場合において、前記繰り返し回数、前記エッジの長さ、又は前記エッジの分岐角度をゲーム状況の変化に応じて変化させることを特徴とする。

【0023】このようにすれば、繰り返し回数、エッジの長さ、エッジの分岐角度を変化させるという簡素な処理で、季節が変化する様子や、樹木が生長する様子などを表現できるようになり、少ない処理負担でよりリアルな画像を生成できる。

【0024】また本発明に係るゲームシステム、情報記憶媒体及びプログラムは、自己相似形を再帰的に生成することで得られる情報に基づいて、オブジェクトを構成する基本パーツを配置する場合において、前記基本パーツをゲーム状況の変化に応じて差し替えることを特徴とする。

【0025】このようにすれば、同一の再帰型構造式を用いながらも、オブジェクトの画像を様々に変化させることが可能になり、プレイヤーの仮想現実感を向上できる。

【0026】また本発明は、画像生成を行うゲームシステムであって、再帰型の構造式と形状特定パラメータとに基づいて自己相似形を再帰的に生成する手段と、自己相似形の再帰的生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像を生成する手段を含み、処理対象となっている再帰型構造式のコードが再帰呼び出しを必要とするコードであった場合には、再帰

呼び出しを実行して呼び出し先の再帰型構造式のコードの処理に移行し、呼び出し先の再帰型構造式の全てのコードの処理が終了した後に、呼び出し元の再帰型構造式のコードの処理に戻ることを特徴とする。また本発明に係る情報記憶媒体は、コンピュータにより使用可能な情報記憶媒体であって、上記手段を実行するためのプログラムを含むことを特徴とする。また本発明に係るプログラムは、コンピュータにより使用可能なプログラム（搬送波に具現化されるプログラムを含む）であって、上記手段を実行するための処理ルーチンを含むことを特徴とする。

【0027】本発明によれば、再帰型構造式と形状特定パラメータとに基づいて自己相似形が再帰的に生成される。そして、この生成により得られた情報に基づいてオブジェクト空間内の所与の視点での画像が生成される。

【0028】そして本発明では、処理対象のコードが再帰呼び出しを必要とするコード（例えば、ノード・リライティングにおけるノードのコード、エッジ・リライティングにおけるエッジのコード）であった場合には、再帰呼び出しが実行され、呼び出し先の構造式のコードの処理に移行する。そして、呼び出し先での全てのコードの処理が終了したことを条件に、呼び出し元の構造式のコードの処理に戻る。このようにすれば、スタックなどに一時的に保持しておかなければならないデータ（例えば位置、回転マトリクス等）の量を低減できるようになり、メモリの使用記憶容量の節約を図れる。

【0029】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて説明する。

【0030】1. 構成

図2に、本実施形態のゲームシステム（画像生成システム）の機能ブロック図の一例を示す。なお同図において本実施形態は、少なくとも処理部100を含めばよく（或いは処理部100と記憶部170を含めばよく）、それ以外のブロックについては任意の構成要素とすることができる。

【0031】操作部160は、プレーヤが操作データを入力するためのものであり、その機能は、レバー、ボタン、マイク、或いは筐体などのハードウェアにより実現できる。

【0032】記憶部170は、処理部100や通信部196などのワーク領域となるもので、その機能はRAMなどのハードウェアにより実現できる。

【0033】情報記憶媒体（コンピュータにより使用可能な記憶媒体）180は、プログラムやデータなどの情報を格納するものであり、その機能は、光ディスク（CD、DVD）、光磁気ディスク（MO）、磁気ディスク、ハードディスク、磁気テープ、或いはメモリ（ROM）などのハードウェアにより実現できる。処理部100は、この情報記憶媒体180に格納される情報に基づ

いて本発明（本実施形態）の種々の処理を行う。即ち情報記憶媒体180には、本発明（本実施形態）の手段（特に処理部100に含まれるブロック）を実行するための情報（プログラム或いはデータ）が格納される。

【0034】なお、情報記憶媒体180に格納される情報の一部又は全部は、システムへの電源投入時等に記憶部170に転送されることになる。また情報記憶媒体180には、本発明の処理を行うためのプログラム、画像データ、音データ、表示物の形状データ、本発明の処理を指示するための情報、或いはその指示に従って処理を行うための情報などを含ませることができる。

【0035】表示部190は、本実施形態により生成された画像を出力するものであり、その機能は、CRT、LCD、或いはHMD（ヘッドマウントディスプレイ）などのハードウェアにより実現できる。

【0036】音出力部192は、本実施形態により生成された音を出力するものであり、その機能は、スピーカなどのハードウェアにより実現できる。

【0037】携帯型情報記憶装置194は、プレーヤの個人データやゲームのセーブデータなどが記憶されるものであり、この携帯型情報記憶装置194としては、メモリカードや携帯型ゲーム装置などを考えることができる。

【0038】通信部196は、外部（例えばホスト装置や他のゲームシステム）との間で通信を行うための各種の制御を行うものであり、その機能は、各種プロセッサ、或いは通信用ASICなどのハードウェアや、プログラムなどにより実現できる。

【0039】なお本発明（本実施形態）の手段を実行するためのプログラム或いはデータは、ホスト装置（サーバー）が有する情報記憶媒体からネットワーク及び通信部196を介して情報記憶媒体180に配信するようにしてもよい。このようなホスト装置（サーバー）の情報記憶媒体の使用も本発明の範囲内に含まれる。

【0040】処理部100（プロセッサ）は、操作部160からの操作データやプログラムなどに基づいて、ゲーム処理、画像生成処理、或いは音生成処理などの各種の処理を行う。この処理部100の機能は、各種プロセッサ（CPU、DSP等）又はASIC（ゲートアレイ等）などのハードウェアや、所与のプログラム（ゲームプログラム）により実現できる。

【0041】なお、処理部100の機能は、その全てをハードウェアにより実現してもよいし、その全てをプログラムにより実現してもよい。或いは、ハードウェアとプログラムの両方により実現してもよい。

【0042】処理部100は、ゲーム処理部110、画像生成部120、音生成部130を含む。

【0043】ここでゲーム処理部110は、コイン（代価）の受け付け処理、各種モードの設定処理、ゲームの進行処理、選択画面の設定処理、オブジェクト（1又は

複数のプリミティブ面)の位置や回転角度(X、Y又はZ軸回り回転角度)を求める処理、オブジェクトを動作させる処理(モーション処理)、視点の位置(仮想カメラの位置)や視線角度(仮想カメラの回転角度)を求める処理、マップオブジェクトなどのオブジェクトをオブジェクト空間へ配置する処理、ヒットチェック処理、ゲーム結果(成果、成績)を演算する処理、複数のプレイヤーが共通のゲーム空間でプレイするための処理、或いはゲームオーバー処理などの種々のゲーム処理を、操作部160からの操作データやプログラムなどに基づいて実行する。

【0044】画像生成部120は、ゲーム処理部110からのゲーム処理結果に基づいて、例えばオブジェクト空間内において所与の視点(仮想カメラ)から見える画像を生成し、表示部190に出力する。

【0045】音生成部130は、ゲーム処理部110からのゲーム処理結果に基づいて各種の音処理を行い、BGM、効果音、又は音声などの音を生成し、音出力部192に出力する。

【0046】ゲーム処理部110は、自己相似形生成部112、距離演算部114、構造式・パラメータ変化部116、パーツ差し替え部118を含む。

【0047】ここで自己相似形生成部112は、自己相似形を再帰的に繰り返し生成する処理を行う。Lシステムの手法を利用する場合を例にとれば、再帰型の構造式と、繰り返し回数、エッジ(枝)の長さ又はエッジの分岐角度などを含む形状特定パラメータとに基づいて、自己相似形を再帰的に生成する処理を行う。

【0048】なお、自己相似形を再帰的に生成することで得られる情報に基づいて画像を生成する場合には、得られた情報に基づいてオブジェクトを構成する基本パーツを配置することで画像を生成してもよいし、得られた情報に基づいてテクスチャを生成し、生成されたテクスチャを利用して画像を生成してもよい。

【0049】距離演算部114は、視点からの距離(視点とオブジェクトの距離)を演算する処理を行う。ここで、視点からの距離としては、オブジェクトの奥行き距離、視点とオブジェクトとの直線距離、或いはこれらの距離と均等な種々のパラメータ(例えば透視変換後のオブジェクトのスクリーン上での大きさ等)を考慮することができる。

【0050】構造式・パラメータ変化部116は、再帰型構造式や形状特定パラメータ(幾何パラメータ)を、距離演算部114で得られた視点からの距離や、ゲーム状況の変化に応じて変化させる処理を行う。より具体的には、形状特定パラメータが含む自己相似形の繰り返し回数を、視点からの距離に応じて変化させたり、ゲーム状況の変化に応じて、再帰型構造式のコードを書き換えたり、再帰型構造式に新たなコードを付加したり、再帰型構造式のコードを削除する。或いは、ゲーム状況の変

化に応じて、自己相似形の繰り返し回数や自己相似形を構成するエッジの長さやエッジの分岐角度を変化させる。

【0051】なおゲーム状況の変化とは、例えば、ゲーム世界において種々のイベント(燃えイベント、季節変化イベント、ゲームステージ変化イベント等)が発生する場合や、ゲーム処理における種々の判断(ゲーム進行の判断、ゲーム結果の判断等)に使用されるパラメータ(時間パラメータ、キャラクタのステータスパラメータ等)が変化する場合などである。

【0052】また再帰型構造式や形状特定パラメータなどの情報は、記憶部170が含む主記憶部172に記憶されることになる。

【0053】パーツ差し替え部118は、オブジェクトを構成する基本パーツをゲーム状況の変化に応じて差し替える処理を行う。即ち、自己相似形の生成により得られる情報に基づいてオブジェクトの基本パーツを配置する場合には、基本パーツを差し替えることで、同一の構造式、形状特定パラメータを用いながらも、異なった印象を受ける画像を生成できるようになる。

【0054】なお、本実施形態のゲームシステムは、1人のプレイヤーのみがプレイできるシングルプレイヤーモード専用のシステムにしてもよいし、このようなシングルプレイヤーモードのみならず、複数のプレイヤーがプレイできるマルチプレイヤーモードも備えるシステムにしてもよい。

【0055】また複数のプレイヤーがプレイする場合に、これらの複数のプレイヤーに提供するゲーム画像やゲーム音を、1つの端末を用いて生成してもよいし、ネットワーク(伝送ライン、通信回線)などで接続された複数の端末(ゲーム機、携帯電話)を用いて生成してもよい。

【0056】2. 本実施形態の特徴

2. 1 Lシステム

まず本実施形態で利用されるLシステムの手法について説明する。

【0057】Lシステムは、フラクタルの一種として位置づけられており、植物の数学的な表現に最適なフラクタル手法である。Lシステムでは、再帰型の構造式(コード列)と、繰り返し回数、エッジ(枝)の長さ、エッジの分岐角度などを含む形状特定パラメータを利用する。再帰型の構造式は、形状特定パラメータが含む繰り返し回数で指定された回数だけ展開される。そして、構造式の展開手法としてはノード・リライティングとエッジ・リライティングがある。

【0058】例えば、下記の構造式(1)をノード・リライティングで展開すると、図3に示すような自己相似形構造の図形が生成される。

【0059】

$A \rightarrow B [+A] B [-A] BA$ (1)

なお、構造式が含む各コードと各コードの動作内容の対

応関係表を図4に示す。"00"はコードの終端であることを指示する。"01(A)"、"02(B)"、"03(L)"、"04(F)"は、各々、ノード、枝、葉、花の生成を指示する。"06(D)"、"07(J)"は分岐の開始、終了を指示する。"08(+)"、"09(-)"は右曲がり、左曲がりを指示する。"0A(&)"、"0B(^)"は、右ピッチ、左ピッチを指示する。"0C(¥)"、"0D(/)"は、右ロール、左ロールを指示する。"0E(|)"は、180度回転を指示する。

【0060】上記の構造式(1)は次のように展開される。

①まず、構造式(1)の最初のコードが"B"になっているため、図5(A)に示すように枝B1を生成する。なお、生成される各枝(広義にはエッジ)の長さは形状特定パラメータにより特定される。

②構造式(1)の次のコードは"[+A]"になっているため、図5(B)に示すように右分岐のノードN1を設定する。なお、各ノードでの分岐角度は形状特定パラメータにより特定される。

③構造式(1)の次のコードは"B"になっているため、図5(C)に示すように枝B2を生成する。

④次のコードは"[-A]"になっているため、図5(D)に示すように左分岐のノードN2を設定する。

⑤次のコードは"BA"になっているため、図5(E)に示すように枝B3を生成すると共にノードN3を設定する。

⑥全てのコードの処理が完了したため、"A→"の規則に従い、枝(エッジ)の長さを半分にした自己相似形を、各ノードN1、N2、N3の位置に再帰的に生成する。即ち、構造式(1)の"B[+A]B[-A]BA"の"A"の部分に、"B[+A]B[-A]BA"を代入して、構造式(1)を再帰的に展開する。

【0061】例えば図5(F)～(J)に示すように、ノードN1の位置では、枝の長さを半分にした右分岐の自己相似形(B11、B12、B13)が生成される。同様に、ノードN2では左分岐の自己相似形が生成され、ノードN3では直進方向の自己相似形が生成される。

【0062】上記の構造式(1)の再帰的な展開を、形状特定パラメータが含む繰り返し回数分だけ繰り返すことで、図3に示すような複雑な自己相似構造の図形を生成できる。

【0063】一方、下記の構造式(2)をエッジ・リライティングで展開すると、図6に示すような自己相似構造の図形が生成される。

【0064】

B→B[+B]B[-B]B (2)

構造式(2)は次のように展開される。

①まず、構造式(2)の最初のコードが"B"になって

いるため、図7(A)に示すように枝B1を生成する。

②構造式(2)の次のコードは"[+B]"になっているため、図7(B)に示すように右分岐の枝B2を生成する。

③構造式(2)の次のコードは"B"になっているため、図7(C)に示すように枝B3を生成する。

④次のコードは"[-B]"になっているため、図7(D)に示すように左分岐の枝B4を生成する。

⑤次のコードは"B"になっているため、図7(E)に示すように枝B5を生成する。

⑥全てのコードの処理が完了したため、"B→"の規則に従い、枝(エッジ)の長さを半分にした自己相似形を、各枝B1、B2、B3、B4、B5の位置に再帰的に生成する。即ち、構造式(2)の"B[+B]B[-B]B"の"B"の部分に、"B[+B]B[-B]B"を代入して、構造式(2)を再帰的に展開する。

【0065】例えば図7(F)～(J)に示すように、枝B1の位置では、枝の長さを半分にした自己相似形(B11、B12、B13、B14、B15)が生成されて上描きされる。同様に、枝B2では右分岐、枝B3では直進方向、枝B4では左分岐、枝B5では直進方向の自己相似形が生成されて上描きされる。

【0066】構造式(2)の再帰的な展開を、形状特定パラメータが含む繰り返し回数分だけ繰り返すことで、図6に示すような複雑な自己相似構造の図形を生成できる。

【0067】図8では、図3、図6のような自己相似形の再帰的生成により得られた情報に基づいて、オブジェクトOB(樹木)を構成する枝、葉、花などの基本パーツ(ポリゴン等のプリミティブ面で構成される)を配置し、オブジェクトOBの画像を生成している。即ち、"B(02)"のコードで生成されたエッジには、その方向に沿って枝パーツが配置され、"L(03)"のコードで生成されたエッジには、その方向に沿って葉パーツが配置され、"F(04)"のコードで生成されたエッジには、その方向に沿って花パーツが配置される。

【0068】このようにすれば、基本パーツを共通に使用しながら、構造式だけを異ならせることで、多様な形状のオブジェクト(樹木)の画像を生成できるようになる。この場合、構造式のデータ量は多くても数百バイト程度であるため、メモリの使用記憶容量も圧迫しない。

【0069】また基本パーツ自体を差し替えば、オブジェクトの画像のバリエーションを容易に増やすことができる。

【0070】例えば基本パーツの形状やテクスチャを変更するだけで、異なった印象のオブジェクトの画像を生成できるようになる。多様な基本パーツを用意しておけば、これらの基本パーツをゲーム状況によって使い分けることも可能となる。そして、基本パーツは枝や葉といった単位なので、数種類登録してもメモリの使用記憶容

量をそれほど圧迫しない。

【0071】なお、図3、図6のように描かれた図形そのものを樹木の画像として採用したり、図3、図6のように描かれた図形をテクスチャとして設定し、このテクスチャを所与のオブジェクトにマッピングすることで所望の画像を生成してもよい。

【0072】2. 2 繰り返し回数の可変制御によるLODの実現

さて、本実施形態の1つの特徴は、視点からの距離に応じて、自己相似形の生成の繰り返し回数を変化させる点にある。

【0073】即ち図9(A)に示すように、視点からの距離(視点とオブジェクトOBの距離)が近い場合には、自己相似形の生成の繰り返し回数(再帰回数)を大きくし、オブジェクトOBの精密度を高める。

【0074】一方、図9(B)に示すように、視点からの距離が中ぐらい場合には、繰り返し回数も中ぐらいにして、オブジェクトOBの精密度を中ぐらいにする。

【0075】また図9(C)に示すように、視点からの距離が遠い場合には、繰り返し回数も少なくして、オブジェクトOBの精密度を低くする。

【0076】例えばLシステムのように、再帰型の構造式と繰り返し回数(形状特定パラメータ)とに基づいて自己相似形を再帰的に生成する場合には、図10に示すように、視点からの距離に応じて変化する繰り返し回数と、同一の再帰型構造式とに基づいて、視点からの距離に応じて精密度が変化するオブジェクトOBの画像を生成する。

【0077】このようにすれば、各距離に応じた適切な精密度でオブジェクトOBの画像を生成でき、少ない処理負担でよりリアルな画像を生成できるようになる。

【0078】なお図9(A)、(B)、(C)、図10では、距離の段階を近距離、中距離、遠距離というように3段階に分けたが、2段階に分けてもよいし、4段階以上に分けてもよい。

【0079】例えば従来のLODにおいては、図1(A)に示すように、オブジェクトを表すモデルとして、近距離用、中距離用、遠距離用モデルというように

$$A \rightarrow [+ \text{ ¥ ¥ } A - - / L] [- / A + + F] + A \quad (3)$$

本実施形態ではゲーム状況の変化に応じて構造式のコード" F" を葉のコード" L" に書き換えると、構造式は

【0087】例えば上記の構造式(3)の中の花のコード

$$A \rightarrow [+ \text{ ¥ ¥ } A - - / L] [- / A + + L] + A \quad (4)$$

この構造式(4)を用いると、図11(A)のオブジェクトOBの画像は図11(B)に示すような画像に変化する。

【0089】また本実施形態ではゲーム状況の変化に応じて構造式にコードを付加する。

$$A \rightarrow [+ \text{ ¥ ¥ } A - - / L] [+ \text{ ¥ ¥ } L] [/ / + L] - / A + + L] + A \quad (5)$$

複数のモデルを用意し、視点からの距離に応じて、使用するモデルを切り替えていた。

【0080】しかしながら、この従来例では図1(B)に示すように、用意するモデルの数の分だけモデルデータが必要になる。従って、モデルデータの使用記憶容量が増大化してしまうという問題点がある。

【0081】また、モデル切り替えの際の画面のショックが大きく、モデルが切り替わったことをプレーヤに気づかれてしまうという問題点もある。そして、この問題点を解決すべく、モデルの数を更に増やすと、モデルデータのデータ量の増大化の問題は更に深刻なものとなる。

【0082】これに対して本実施形態では図10に示すように、精密度の異なるオブジェクトOBの画像を、同一の構造式を用いて生成できる。即ち、同一の構造式と同一の基本パーツを用いながらも、繰り返し回数だけを距離に応じて変化させるだけで、視点からの距離に応じてオブジェクトOBの精密度を変化させることができる。従って、多数のモデルを用意することなくLODを実現できるようになり、メモリの使用記憶容量を節約できる。

【0083】また本実施形態によれば、繰り返し回数シームレスに変化させることでオブジェクトの精密度をシームレスに変化させることも可能となり、画面に生じるショック等を防止することも可能になる。また、このように繰り返し回数をシームレスに変化させた場合にも、構造式や基本パーツのデータ量は不変であるため、図1(B)に示すようなモデルデータのデータ量が極端に増加してしまうという問題も生じない。

【0084】2. 3 ゲーム状況に応じた構造式、形状特定パラメータ、基本パーツの変化さて本実施形態では、再帰型の構造式や形状特定パラメータをゲーム状況の変化に応じてリアルタイムに変化させている。

【0085】例えば原形となる構造式が下式(3)であった場合には、図11(A)に示すようなオブジェクトOBの画像が生成される。

【0086】

下式(4)のようになる。

【0088】

【0090】例えば上記の構造式(4)にコード" [+ ¥ ¥ L] [/ / + L]" を付加すると、構造式は下式(5)のようになる。

【0091】

この構造式(5)を用いると、図11(B)のオブジェクトOBの画像は図11(C)に示すような画像に変化する。

【0092】なお、ゲーム状況の変化に応じて構造式のコードを削除するようにしてよい。

【0093】例えば、ゲーム世界での季節が変化した場合(広義にはゲーム状況が変化した場合)に、このようなコードの書き換え、付加、又は削除を行えば、オブジェクトOBの画像を各季節に応じた適切な画像にすることができ、画像のリアルさを増すことができる。しかも、必要な処理は構造式のコードの書き換えや付加や削除だけで済むため、処理負荷も軽い。

【0094】また図11(A)、(B)、(C)のような画像変化をモデル差し替えの手法により実現しようとすると、異なった複数種類のポリゴンモデルが必要になってしまい、メモリの使用記憶容量を圧迫してしまう。

【0095】これに対して本実施形態では、図11(A)、(B)、(C)の画像を生成するための原形となる構造式は1つで済み、構造式のデータ量も数百バイト程度と少ない。従って、モデル差し替えの手法に比べて、メモリの使用記憶容量を大幅に節約できる。

【0096】図12では、ゲーム状況の変化に応じて、形状特定パラメータが含む繰り返し回数を変化させている。このように繰り返し回数を変化させることで、オブジェクトOBの形状の複雑さをゲーム状況に応じて段階的に調節できるようになる。

【0097】例えば、時間パラメータが変化し、ゲーム世界での時間が経過するにつれて、繰り返し回数を増やすようにすれば、樹木が徐々に生長して行く様子などを表現できる。

【0098】また、他のオブジェクトに対する処理の負荷が重くなったために図2の処理部100(CPU)の処理に余裕が無くなるなどのゲーム状況の変化が生じた場合には、繰り返し回数を少なくすることで、処理部100の処理に余裕を持たせることが可能になる。

【0099】なお、ゲーム状況の変化に応じて、形状特定パラメータが含むエッジの長さやエッジの分岐角度を変化させるようにしてもよい。例えば、エッジ(枝)の長さを時間パラメータの変化に応じて変化させることで、樹木が徐々に生長して行く様子などを表現できる。

【0100】また本実施形態では、ゲーム状況の変化に応じて、オブジェクトを構成する基本パーツを差し替えるようにしている。

【0101】例えば図13(A)、(B)では、オブジェクトOBを構成する枝パーツをゲーム状況の変化に応じて差し替えている。即ち、図13(A)では白のテクスチャがマッピングされた枝パーツを使用しており、図13(B)では茶色のテクスチャがマッピングされた枝パーツを使用している。

【0102】例えばゲーム世界の季節が冬である場合に

は図13(A)に示すような白のテクスチャがマッピングされた枝パーツを使用し、ゲーム世界の季節が夏である場合には図13(B)に示すような茶色のテクスチャがマッピングされた枝パーツを使用するようにする。このようにすれば、季節の変化に応じてオブジェクトOBの画像が変化するようになり、プレイヤーの仮想現実感を増すことができる。

【0103】3. 本実施形態の処理

次に、本実施形態の処理の詳細例について、図14～図19のフローチャートを用いて説明する。

【0104】図14は、メインルーチンの処理のフローチャートである。

【0105】まず、処理の初期化を行う(ステップS1)。より具体的には、原形となる構造式や、形状特定パラメータや、基本パーツを用意する処理などを行う。

【0106】次に、樹木の燃えイベントが発生したか否か(広義にはゲームイベントが発生したか否か)について判断する(ステップS2)。例えばフライトシミュレーションゲームにおいて、マップ上の森に飛行機が墜落したりミサイルが命中すると、樹木の燃えイベントが発生する。そして燃えイベントが発生したと判断されると、燃えイベント処理に移行する(ステップS3)。

【0107】次に、季節変化の処理を行う(ステップS4)。そして、距離変化の処理(ステップS5)、時間経過の処理(ステップS6)を行い、ステップS2に戻る。これらの処理は例えば各フレーム毎に行われる。

【0108】図15は、燃えイベント処理のフローチャートである。

【0109】飛行機の墜落やミサイルの命中などにより樹木に火が燃え移ったと判断されると(ステップS11)、樹木を表す構造式が含む葉のコード、花のコードを削除する(ステップS12)。そして、枝パーツを枯れ枝を表すパーツに差し替える(ステップS13)。

【0110】このようにすれば、樹木は枝パーツのみにより構成されるようになり、しかもその枝パーツも枯れ枝を表すパーツに差し替えられる。従って、あたかも樹木が火により燃えてしまったかのように見せることができ、プレイヤーの仮想現実感を増すことができる。

【0111】図16は、季節変化の処理のフローチャートである。

【0112】まず、ゲーム世界での現在の季節がどの季節であるかをチェックする(ステップS21)。そして、現在の季節が春であった場合には、樹木を表す構造式につぼみのコードを付加する(ステップS22、S23)。また、現在の季節が夏であった場合には、つぼみのコードを花のコードに書き換える(ステップS24、S25)。また現在の季節が秋であった場合には、花のコードを実のコードに書き換える(ステップS26、S27)。また現在の季節が冬であった場合には、実のコードを構造式から削除する(ステップS28、S2

9)。

【0113】このようにすることで、春になると樹木につぼみがつき、夏になるとそのつぼみが花になり、秋になるとその花が実になり、冬になるとその実が落ちてなくなるといような表現が可能になる。これによって、プレーヤに季節の変化を感じさせることが可能になり、プレーヤの仮想現実感を増すことができる。

【0114】図17は、距離変化の処理(LOD)のフローチャートである。

【0115】まず、視点からオブジェクトまでの距離Lを求める(ステップS31)。この場合、透視変換後のオブジェクトのスクリーン上での大きさを距離Lのパラメータとして設定するようにしてもよい。

【0116】次に、距離Lがnear(近距離)よりも小さいか否かを判断する(ステップS32)。そして、小さい場合には、繰り返し回数Rにrepeat(繰り返し回数の初期値)をセットする(ステップS33)。

【0117】次に、距離Lがmiddle(中距離)よりも小さいか否かを判断する(ステップS34)。そして、小さい場合には、繰り返し回数Rにrepeat-1をセットする(ステップS35)。一方、距離Lがmiddleよりも大きい場合には、繰り返し回数Rにrepeat-2をセットする(ステップS35)。

【0118】このようにすれば、視点からオブジェクトまでの距離Lが大きいほど繰り返し回数Rが小さくなり(Lが小さいほどRが大きくなり)、視点からの距離に応じてオブジェクトの精密度(形状の複雑さ)を変化させることができる。

【0119】図18は時間経過処理のフローチャートである。

【0120】まず、生長度growが生長限界maxよりも小さいか否かを判断する(ステップS41)。そして、小さい場合には、生長度growを1だけインクリメントする(ステップS42)。そして、枝(エッジ)の長さlengthを、 $length = grow \times weight$ (重み付け係数)の式に従って求める(ステップS43)。

【0121】このようにすれば、時間経過に伴い枝の長さlengthが伸びて行くようになり、樹木が生長して行く様子をリアルに表現できる。

【0122】図19は構造式の展開処理のフローチャートである。

【0123】まず、構造式のcode[n]の引数であるnを0にリセットする(ステップS51)。

【0124】次に、codeの終了か否かを判断し(ステップS52)、codeの終了である場合には構造式の展開処理を終了する。例えば、構造式を含むcodeの最大個数が100個であった場合には、100個目のcodeを処理するとcodeの終了であると判断される。

【0125】次に、code[n]が"00(コード終端)"か否かを判断する(ステップS53)。そして、"00"の場合には、再帰の呼び出し先から呼び出し元に処理を戻す(ステップS54)。

【0126】次に、code[n]が"01(ノード)"か否かを判断する(ステップS55)。そして、code[n]が"01"の場合には、count(再帰の深さ、繰り返し回数)が0よりも大きいか否かを判断する(ステップS56)。そして、0よりも大きい場合には、countを1だけデクリメントして(ステップS57)、再帰呼び出しを実行する(ステップS58)。即ち、自分自身の処理ルーチン(構造式の展開処理)を呼び出して、ステップS51に戻る。そして、ステップS54で再帰の呼び出し先から戻ると、countを1だけインクリメントする(ステップS59)。

【0127】一方、ステップS56でcountが0以下と判断された場合には、nを1だけインクリメントする(ステップS69)。

【0128】ステップS55で"N"と判断された場合には、code[n]が"02(枝)、03(葉)、04(花)、05"か否かを判断する(ステップS60)。そして、code[n]が"02~05"の場合には、該当するパーツを描画する(ステップS61)。そして、次の始点となる位置XYZを算出し(ステップS62)、nを1だけインクリメントする(ステップS69)。

【0129】ステップS60で"N"と判断された場合には、code[n]が"06(分岐開始)"か否かを判断する(ステップS63)。code[n]が"06"の場合には、その時の位置XYZ、回転マトリクスMをスタックに積み(ステップS64)、nを1だけインクリメントする(ステップS69)。

【0130】ステップS63で"N"と判断された場合には、code[n]が"07(分岐終了)"か否かを判断する(ステップS65)。code[n]が"07"の場合には、位置XYZ、回転マトリクスMをスタックから取り出し(ステップS66)、nを1だけインクリメントする(ステップS69)。

【0131】ステップS65で"N"と判断された場合には、code[n]が"08(右曲がり)~0E(180度回転)"か否かを判断する(ステップS67)。code[n]が"08~0E"の場合には、回転マトリクスMを求める処理を行い、(ステップS68)、nを1だけインクリメントする(ステップS69)。

【0132】下記の構造式(6)をノード・リライティングで展開した場合の図19の処理について、図20(A)~(H)、図21(A)~(D)を用いて説明する。なお、count(再帰の深さ)の初期値は1であるとする。

【0133】

$A \rightarrow B [+A] B [-A] BA$ (6)

まず、図19のステップS60で $code[0] = 02$ (B)と判断され、ステップS61、S62に移行し、図20(A)に示すように枝B1が生成される。そして、ステップS69に移行し、 $n = 1$ にインクリメントされる。

【0134】次に、ステップS63で $code[1] = 06$ (D)と判断され、ステップS64に移行し、その時の位置XYZ、回転マトリクスMがスタックに積まれる。そして、ステップS69に移行し、 $n = 2$ にインクリメントされる。

【0135】次に、ステップS67で $code[2] = 08$ (+)と判断され、右分岐の回転マトリクスMの処理が行われる。そして、ステップS69に移行し、 $n = 3$ にインクリメントされる。

【0136】次に、ステップS55で $code[3] = 01$ (A)と判断されステップS56に移行する。そして、ステップS56で $count = 1 > 0$ と判断され、ステップS57で $count = 0$ にデクリメントされる。そして、ステップS58で図20(B)に示すように再帰呼び出しが実行され、呼び出し先の構造式(再帰の下位層の構造式)のコードの処理に移行する。即ちステップS51に戻り、 $n = 0$ にリセットされる。

【0137】次に、図20(C)、(D)に示すように、 n が順次インクリメントされながら、呼び出し先の構造式のコードにより枝B11、B12、B13が順次生成される。そして、ステップS53で $code[12] = 00$ (コード終端)と判断され、ステップS54で呼び出し元の構造式(再帰の上位層の構造式)のコードの処理に戻り、ステップS59で $count = 1$ にインクリメントされる。

【0138】次に、図20(E)、(F)に示すように呼び出し元の構造式のコードにより枝B2が生成された後、ステップS58で再帰呼び出しが実行される。そして、図20(G)、(H)に示すように、呼び出し先の構造式のコードにより枝B21、B22、B23が順次生成される。そして、ステップS53で $code[12] = 00$ (コード終端)と判断されると、ステップS54で呼び出し元の構造式のコードの処理に戻る。

【0139】次に、図21(A)、(B)に示すように呼び出し元の構造式のコードにより枝B3が生成された後、ステップS58で再帰呼び出しが実行される。そして、図21(C)、(D)に示すように、呼び出し先の構造式のコードにより枝B31、B32、B33が順次生成される。そして、ステップS53で $code[12] = 00$ (コード終端)と判断されると、ステップS54で呼び出し元の構造式のコードの処理に戻る。

【0140】このように本実施形態では、処理対象となっている構造式のコードが再帰呼び出しを必要とするコード(A)であった場合には、ステップS58で再帰呼

び出しが実行され、呼び出し先の構造式のコードの処理に移行する。

【0141】即ち、図20(B)で再帰呼び出しが実行され、呼び出し先の構造式のコードの処理に移行する。そして、この呼び出し先の構造式のコードの処理により、図20(C)、(D)に示すように枝B11、B12、B13が生成される。

【0142】そして、図20(D)に示すように呼び出し先の構造式の全てのコードの処理が終了すると、ステップS54で呼び出し元の構造式のコードの処理に戻る。即ち、呼び出し元の構造式のコードの処理により図20(E)、(F)に示すように枝B2が生成される。

【0143】このように本実施形態では、再帰呼び出しが実行されると、呼び出し先の構造式の全てのコードの処理を終了させてしまう(図20(D)、図20(H)、図21(D))。そして、呼び出し先の構造式の全てのコードの処理が終了した後に、呼び出し元の構造式のコードの処理に戻る(図20(E)、図21(A))。

【0144】このようにすることで、図5(A)～(J)に示す処理手順に比べて、スタックなどに一時的に保持しておかなければならないデータの量を低減でき、メモリなどのリソースを有効利用できるようになる。

【0145】4. ハードウェア構成

次に、本実施形態を実現できるハードウェアの構成の一例について図22を用いて説明する。

【0146】メインプロセッサ900は、CD982(情報記憶媒体)に格納されたプログラム、通信インターフェース990を介して転送されたプログラム、或いはROM950(情報記憶媒体の1つ)に格納されたプログラムなどに基づき動作し、ゲーム処理、画像処理、音処理などの種々の処理を実行する。

【0147】コプロセッサ902は、メインプロセッサ900の処理を補助するものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算(ベクトル演算)を高速に実行する。例えば、オブジェクトを移動させたり動作(モーション)させるための物理シミュレーションに、マトリクス演算などの処理が必要な場合には、メインプロセッサ900上で動作するプログラムが、その処理をコプロセッサ902に指示(依頼)する。

【0148】ジオメトリプロセッサ904は、座標変換、透視変換、光源計算、曲面生成などのジオメトリ処理を行うものであり、高速並列演算が可能な積和算器や除算器を有し、マトリクス演算(ベクトル演算)を高速に実行する。例えば、座標変換、透視変換、光源計算などの処理を行う場合には、メインプロセッサ900で動作するプログラムが、その処理をジオメトリプロセッサ904に指示する。

【0149】データ伸張プロセッサ906は、圧縮された画像データや音データを伸張するデコード処理を行ったり、メインプロセッサ900のデコード処理をアクセレートする処理を行う。これにより、オープニング画面、インターミッション画面、エンディング画面、或いはゲーム画面などにおいて、MPEG方式等で圧縮された動画像を表示できるようになる。なお、デコード処理の対象となる画像データや音データは、ROM950、CD982に格納されたり、或いは通信インターフェース990を介して外部から転送される。

【0150】描画プロセッサ910は、ポリゴンや曲面などのプリミティブ面で構成されるオブジェクトの描画（レンダリング）処理を高速に実行するものである。オブジェクトの描画の際には、メインプロセッサ900は、DMAコントローラ970の機能を利用して、オブジェクトデータを描画プロセッサ910に渡すと共に、必要であればテクスチャ記憶部924にテクスチャを転送する。すると、描画プロセッサ910は、これらのオブジェクトデータやテクスチャに基づいて、Zバッファなどを利用した陰面消去を行いながら、オブジェクトをフレームバッファ922に高速に描画する。また、描画プロセッサ910は、 α ブレンディング（半透明処理）、デプスキューイング、ミップマッピング、フォグ処理、バイリニア・フィルタリング、トライリニア・フィルタリング、アンチエイリアシング、シェーディング処理なども行うことができる。そして、1フレーム分の画像がフレームバッファ922に書き込まれると、その画像はディスプレイ912に表示される。

【0151】サウンドプロセッサ930は、多チャンネルのADPCM音源などを内蔵し、BGM、効果音、音声などの高品質のゲーム音を生成する。生成されたゲーム音は、スピーカ932から出力される。

【0152】ゲームコントローラ942からの操作データや、メモリカード944からのセーブデータ、個人データは、シリアルインターフェース940を介してデータ転送される。

【0153】ROM950にはシステムプログラムなどが格納される。なお、業務用ゲームシステムの場合には、ROM950が情報記憶媒体として機能し、ROM950に各種プログラムが格納されることになる。なお、ROM950の代わりにハードディスクを利用するようにしてもよい。

【0154】RAM960は、各種プロセッサの作業領域として用いられる。

【0155】DMAコントローラ970は、プロセッサ、メモリ（RAM、VRAM、ROM等）間でのDMA転送を制御するものである。

【0156】CDドライブ980は、プログラム、画像データ、或いは音データなどが格納されるCD982

（情報記憶媒体）を駆動し、これらのプログラム、デー

タへのアクセスを可能にする。

【0157】通信インターフェース990は、ネットワークを介して外部との間でデータ転送を行うためのインターフェースである。この場合に、通信インターフェース990に接続されるネットワークとしては、通信回線（アナログ電話回線、ISDN）、高速シリアルバスなどを考えることができる。そして、通信回線を利用することでインターネットを介したデータ転送が可能になる。また、高速シリアルバスを利用することで、他のゲームシステムとの間でのデータ転送が可能になる。

【0158】なお、本発明の各手段は、その全てを、ハードウェアのみにより実行してもよいし、情報記憶媒体に格納されるプログラムや通信インターフェースを介して配信されるプログラムのみにより実行してもよい。或いは、ハードウェアとプログラムの両方により実行してもよい。

【0159】そして、本発明の各手段をハードウェアとプログラムの両方により実行する場合には、情報記憶媒体には、本発明の各手段をハードウェアを利用して実行するためのプログラムが格納されることになる。より具体的には、上記プログラムが、ハードウェアである各プロセッサ902、904、906、910、930等に処理を指示すると共に、必要であればデータを渡す。そして、各プロセッサ902、904、906、910、930等は、その指示と渡されたデータとに基づいて、本発明の各手段を実行することになる。

【0160】図23（A）に、本実施形態を業務用ゲームシステムに適用した場合の例を示す。プレーヤは、ディスプレイ1100上に映し出されたゲーム画像を見ながら、レバー1102、ボタン1104等を操作してゲームを楽しむ。内蔵されるシステムボード（サーキットボード）1106には、各種プロセッサ、各種メモリなどが実装される。そして、本発明の各手段を実行するための情報（プログラム又はデータ）は、システムボード1106上の情報記憶媒体であるメモリ1108に格納される。以下、この情報を格納情報と呼ぶ。

【0161】図23（B）に、本実施形態を家庭用のゲームシステムに適用した場合の例を示す。プレーヤはディスプレイ1200に映し出されたゲーム画像を見ながら、ゲームコントローラ1202、1204を操作してゲームを楽しむ。この場合、上記格納情報は、本体システムに着脱自在な情報記憶媒体であるCD1206、或いはメモリカード1208、1209等に格納されている。

【0162】図23（C）に、ホスト装置1300と、このホスト装置1300とネットワーク1302（LANのような小規模ネットワークや、インターネットのような広域ネットワーク等）を介して接続される端末1304-1～1304-n（ゲーム機、携帯電話等）とを含むシステムに本実施形態を適用した場合の例を示す。この

10

20

30

40

50

場合、上記格納情報は、例えばホスト装置1300が制御可能な磁気ディスク装置、磁気テープ装置、メモリ等の情報記憶媒体1306に格納されている。端末1304-1~1304-nが、スタンドアロンでゲーム画像、ゲーム音を生成できるものである場合には、ホスト装置1300からは、ゲーム画像、ゲーム音を生成するためのゲームプログラム等が端末1304-1~1304-nに配送される。一方、スタンドアロンで生成できない場合には、ホスト装置1300がゲーム画像、ゲーム音を生成し、これを端末1304-1~1304-nに伝送し端末において出力することになる。

【0163】なお、図23(C)の構成の場合に、本発明の各手段を、ホスト装置(サーバー)と端末とで分散して実行するようにしてもよい。また、本発明の各手段を実行するための上記格納情報を、ホスト装置(サーバー)の情報記憶媒体と端末の情報記憶媒体に分散して格納するようにしてもよい。

【0164】またネットワークに接続する端末は、家庭用ゲームシステムであってもよいし業務用ゲームシステムであってもよい。そして、業務用ゲームシステムをネットワークに接続する場合には、業務用ゲームシステムとの間で情報のやり取りが可能であると共に家庭用ゲームシステムとの間でも情報のやり取りが可能なセーブ用情報記憶装置(メモリカード、携帯型ゲーム装置)を用いることが望ましい。

【0165】なお本発明は、上記実施形態で説明したものに限らず、種々の変形実施が可能である。

【0166】例えば、本発明のうち従属請求項に係る発明においては、従属先の請求項の構成要件の一部を省略する構成とすることもできる。また、本発明の1の独立請求項に係る発明の要部を、他の独立請求項に従属させることもできる。

【0167】また本実施形態では、自己相似形の再帰的生成手法としてLシステムを主に例にとり説明した。本発明における自己相似形の再帰的生成手法はLシステムであることが特に望ましいが、Lシステム以外にも、コッホ(koch)曲線、ドラゴン曲線、中点変位法、反復関数システム(IFS)などを用いることも可能である。

【0168】また本実施形態では樹木の画像を生成する場合について主に例をとり説明した。しかしながら、本発明は、樹木以外の植物や、地形(山岳)や、ガス状物体(雲)などの種々の画像の生成に適用可能である。

【0169】また、構造式の構成や形状特定パラメータの種類についても、種々の変形実施が可能である。

【0170】また、本発明は種々のゲーム(格闘ゲーム、シューティングゲーム、ロボット対戦ゲーム、スポーツゲーム、競争ゲーム、ロールプレイングゲーム、音楽演奏ゲーム、ダンスゲーム等)に適用できる。

【0171】また本発明は、業務用ゲームシステム、家

庭用ゲームシステム、多数のプレーヤが参加する大型アトラクションシステム、シミュレータ、マルチメディア端末、ゲーム画像を生成するシステムボード等の種々のゲームシステム(画像生成システム)に適用できる。

【図面の簡単な説明】

【図1】図1(A)、(B)は、従来のLODの問題点について説明するための図である。

【図2】本実施形態のゲームシステムの機能ブロック図の例である。

【図3】ノード・リライティングについて説明するための図である。

【図4】コードとその動作内容の対応関係表の例である。

【図5】図5(A)~(J)も、ノード・リライティングについて説明するための図である。

【図6】エッジ・リライティングについて説明するための図である。

【図7】図7(A)~(J)も、エッジ・リライティングについて説明するための図である。

【図8】本実施形態により生成される樹木(オブジェクト)の画像の例である。

【図9】図9(A)、(B)、(C)は、視点からの距離に応じて自己相似形の生成の繰り返し回数を変化させる手法について説明するための図である。

【図10】視点からの距離に応じて自己相似形の生成の繰り返し回数を変化させる手法について説明するための図である。

【図11】図11(A)、(B)、(C)は、ゲーム状況の変化に応じて構造式を変化させる手法について説明するための図である。

【図12】ゲーム状況の変化に応じて形状特定パラメータ(繰り返し回数)を変化させる手法について説明するための図である。

【図13】図13(A)、(B)は、ゲーム状況の変化に応じて基本パーツを差し替える手法について説明するための図である。

【図14】本実施形態の処理の詳細例について示すフローチャートである。

【図15】本実施形態の処理の詳細例について示すフローチャートである。

【図16】本実施形態の処理の詳細例について示すフローチャートである。

【図17】本実施形態の処理の詳細例について示すフローチャートである。

【図18】本実施形態の処理の詳細例について示すフローチャートである。

【図19】本実施形態の処理の詳細例について示すフローチャートである。

【図20】図20(A)~(H)は、構造式の展開手法について説明するための図である。

【図21】図21(A)～(D)も、構造式の展開手法について説明するための図である。

【図22】本実施形態を実現できるハードウェアの構成の一例を示す図である。

【図23】図23(A)、(B)、(C)は、本実施形態が適用される種々の形態のシステムの例を示す図である。

【符号の説明】

OB オブジェクト

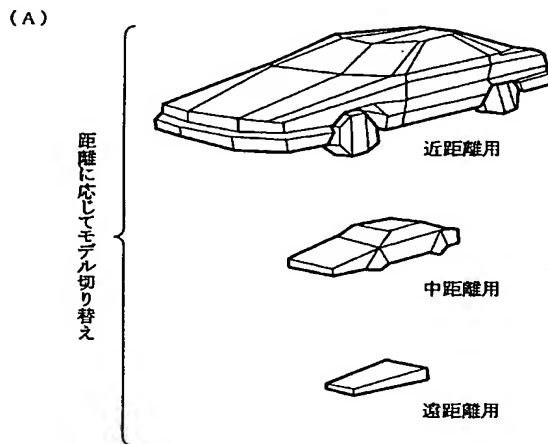
100 処理部

110 ゲーム処理部

112 自己相似形生成部

114 距離演算部

【図1】



(B)

モデルデータ

近距離用モデルデータ					
X ₀	Y ₀	Z ₀	TX ₀	TY ₀	-----
X ₁	Y ₁	Z ₁	TX ₁	TY ₁	-----
中距離用モデルデータ					
X ₀	Y ₀	Z ₀	TX ₀	TY ₀	-----
X ₁	Y ₁	Z ₁	TX ₁	TY ₁	-----
遠距離用モデルデータ					
X ₀	Y ₀	Z ₀	TX ₀	TY ₀	-----
X ₁	Y ₁	Z ₁	TX ₁	TY ₁	-----

116 構造式・パラメータ変化部

118 パーツ差し替え部

120 画像生成部

130 音生成部

160 操作部

170 記憶部

172 主記憶部

180 情報記憶媒体

190 表示部

10 192 音出力部

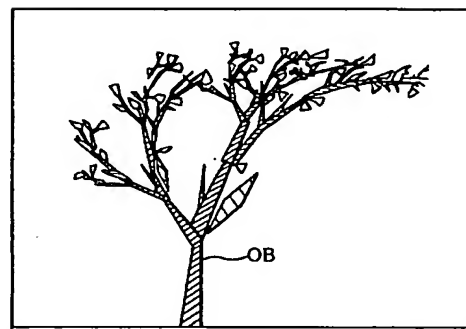
194 携帯型情報記憶装置

196 通信部

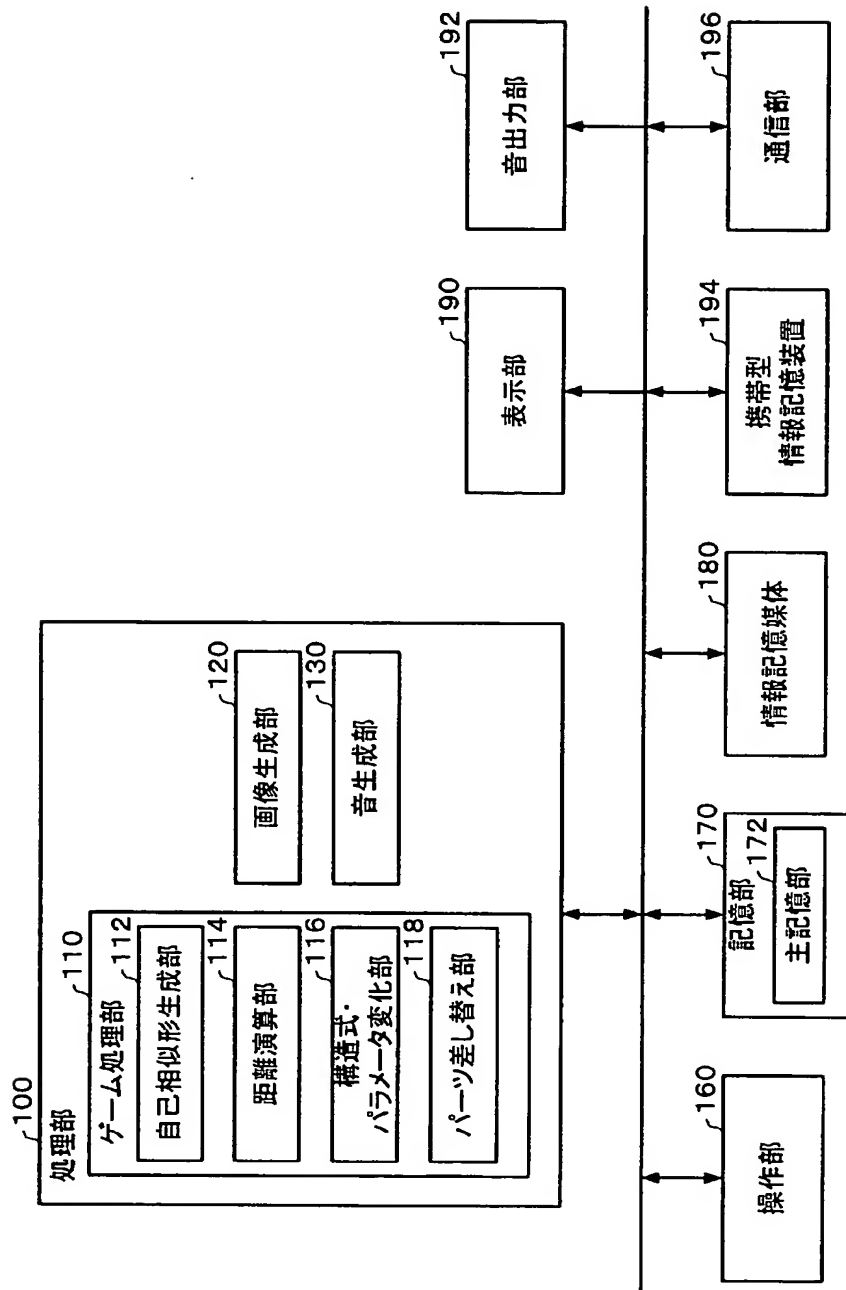
【図4】

コード	記号	動作内容
00		コードの終端
01	A	ノードを生成
02	B	枝を生成
03	L	葉を生成
04	F	花を生成
05		
06	[分岐開始
07]	分岐終了
08	+	右に曲がる
09	-	左に曲がる
0A	&	右にピッチする
0B	^	左にピッチする
0C	¥	右にロールする
0D	/	左にロールする
0E		180度回転
0F		

【図8】

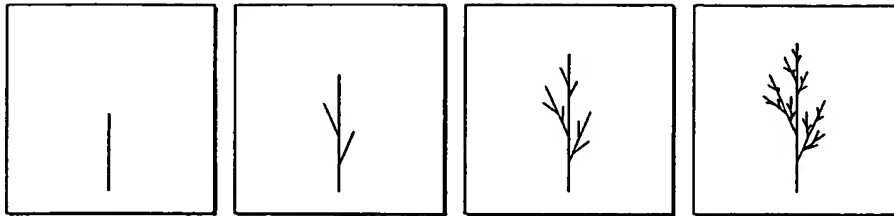


【図2】



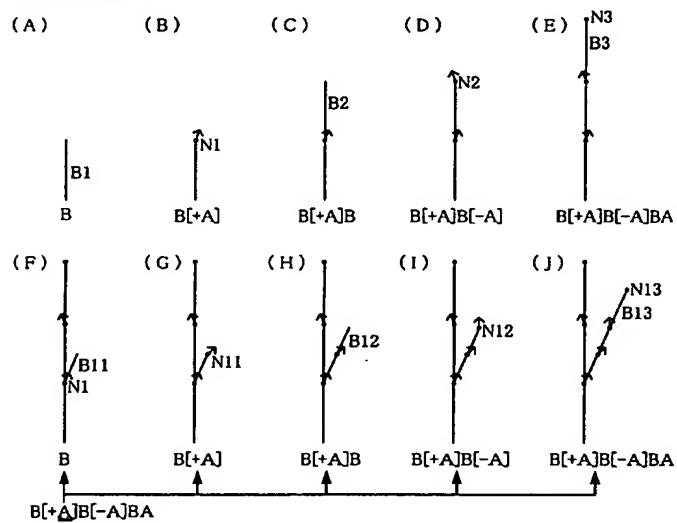
【図3】

ノード・リライティング

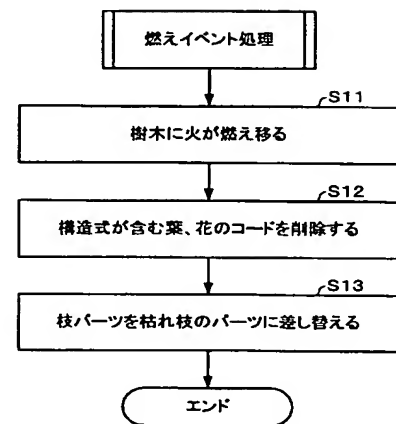


【図5】

ノード・リライティング

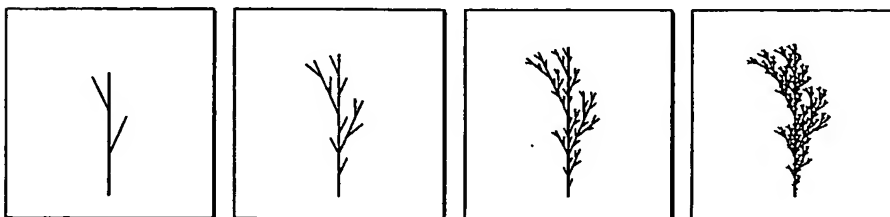


【図15】



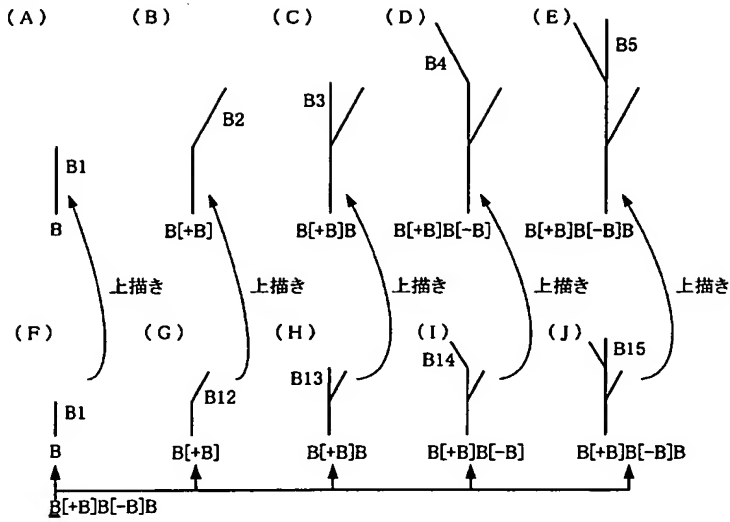
【図6】

エッジ・リライティング

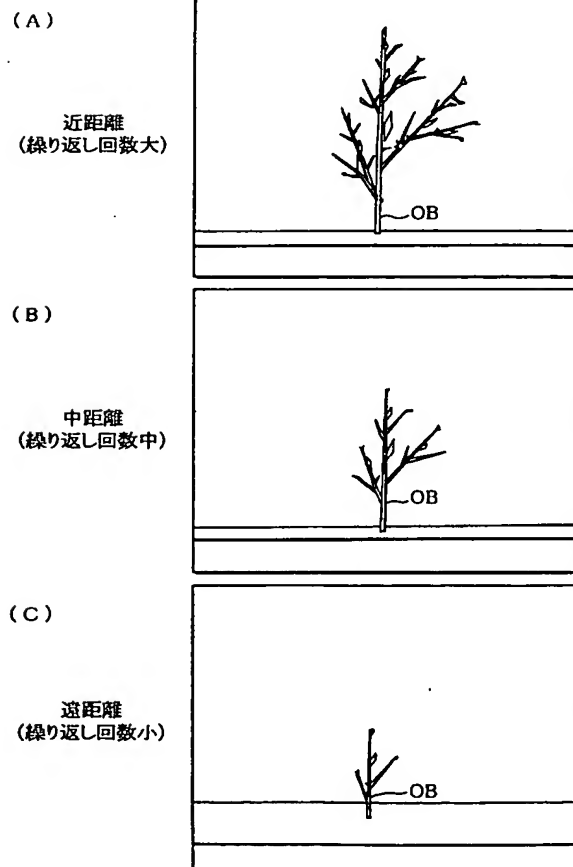


【図7】

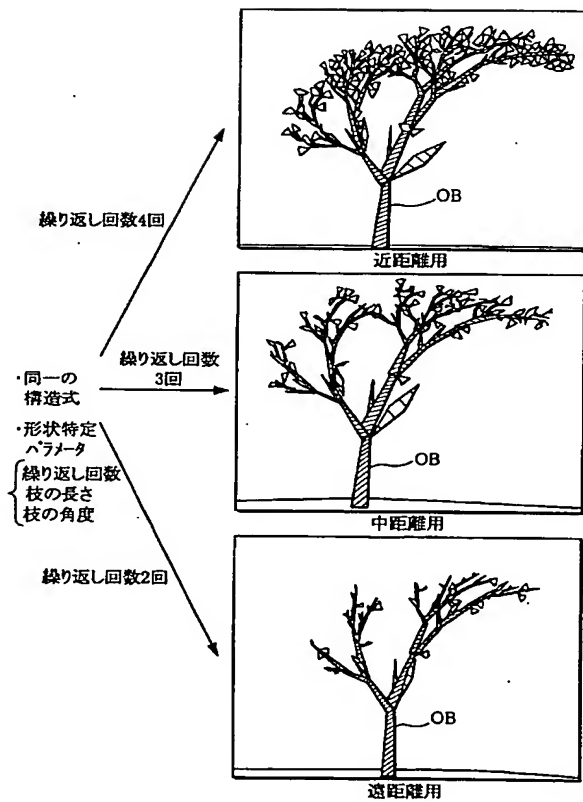
エッジ・リライティング



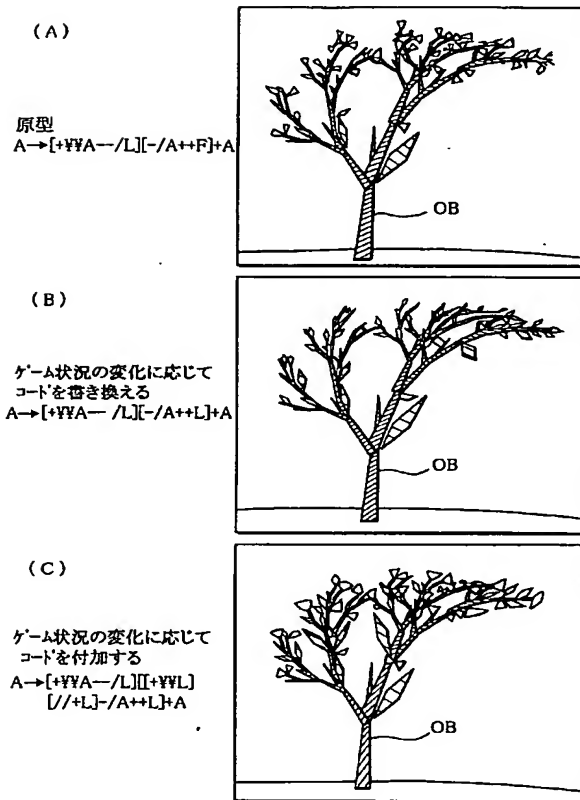
【図9】



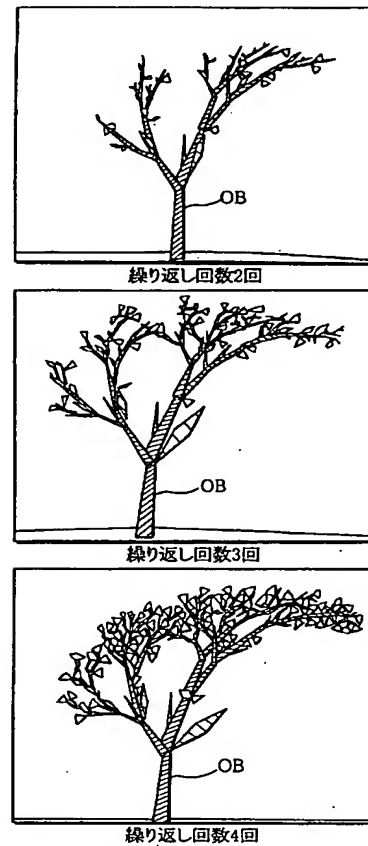
【図10】



【図11】

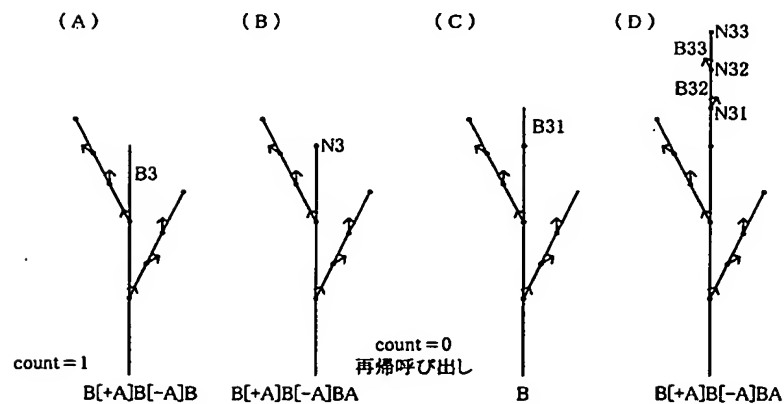


【図12】

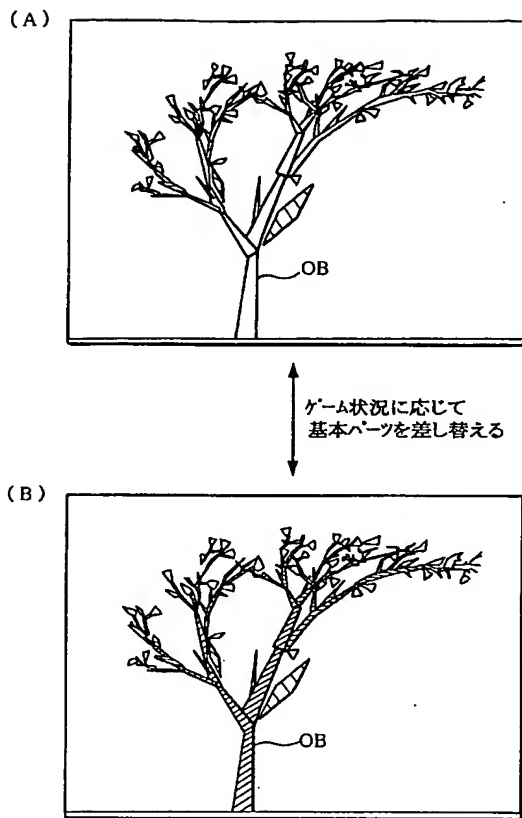


ゲーム状況の
 変化に応じて繰
 り返し回数を変
 化させる

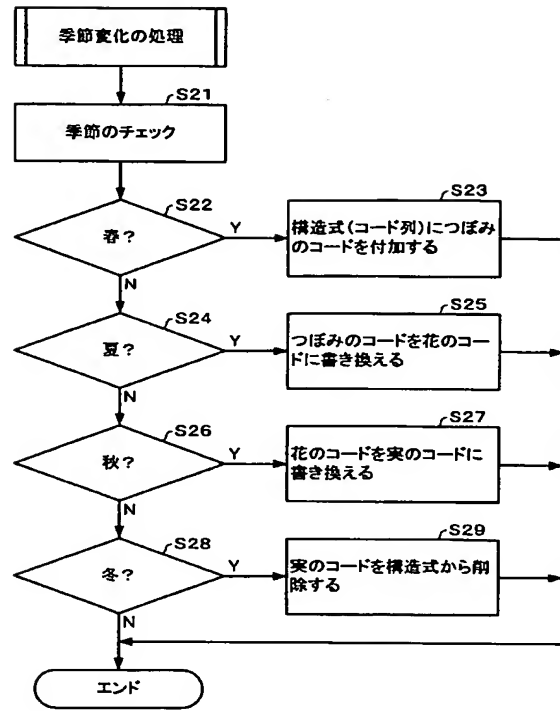
【図21】



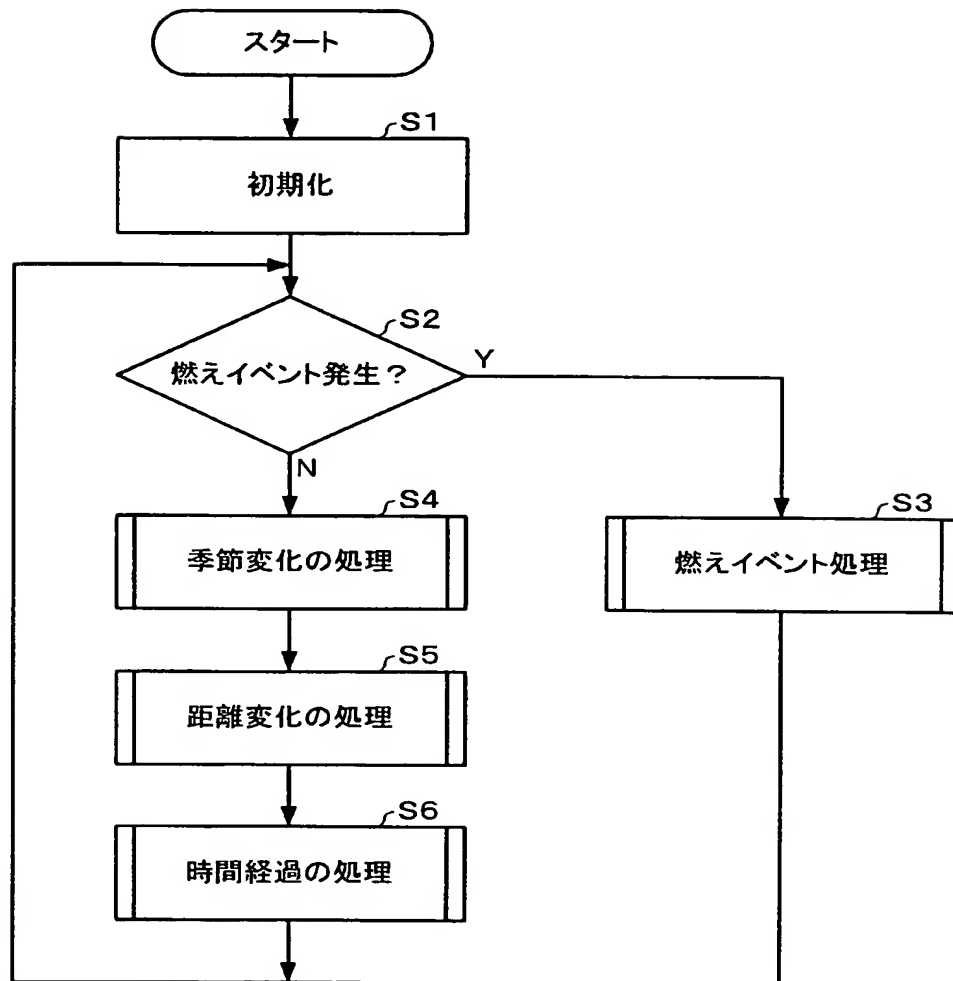
【図13】



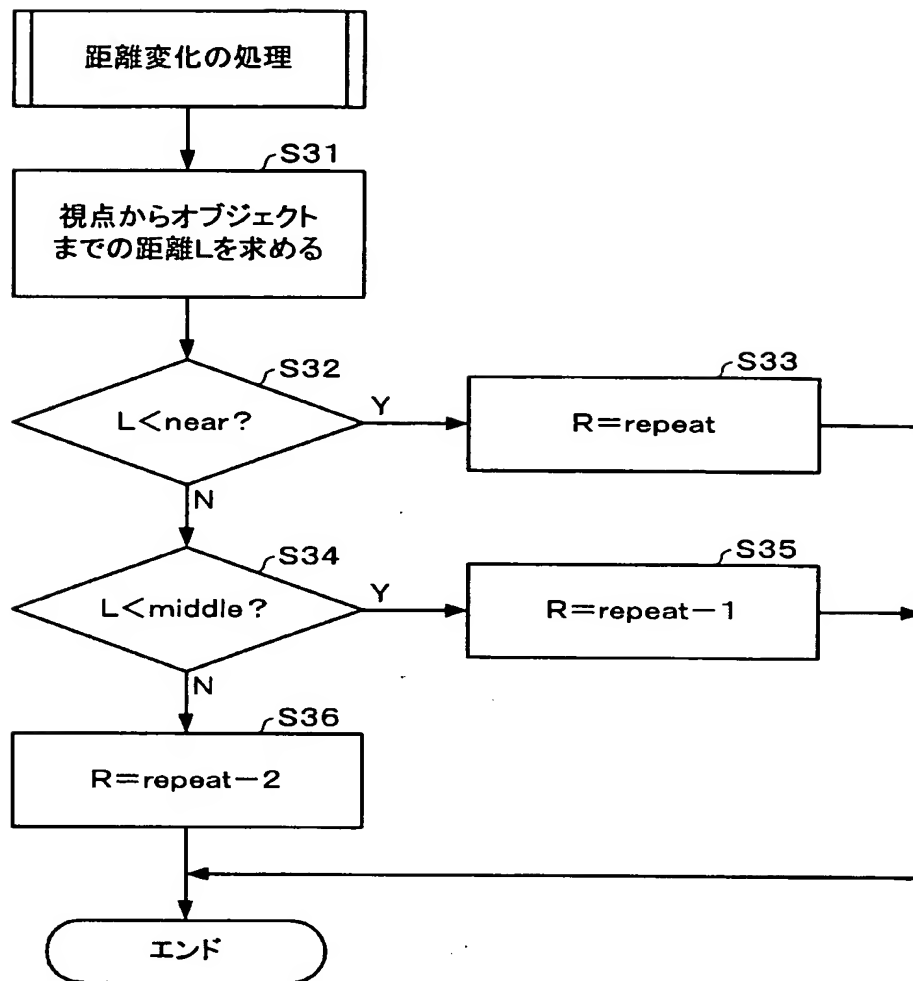
【図16】



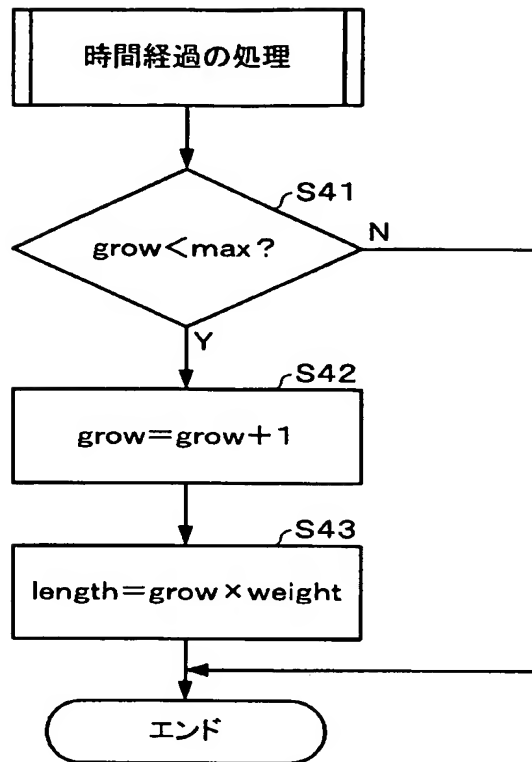
【図14】



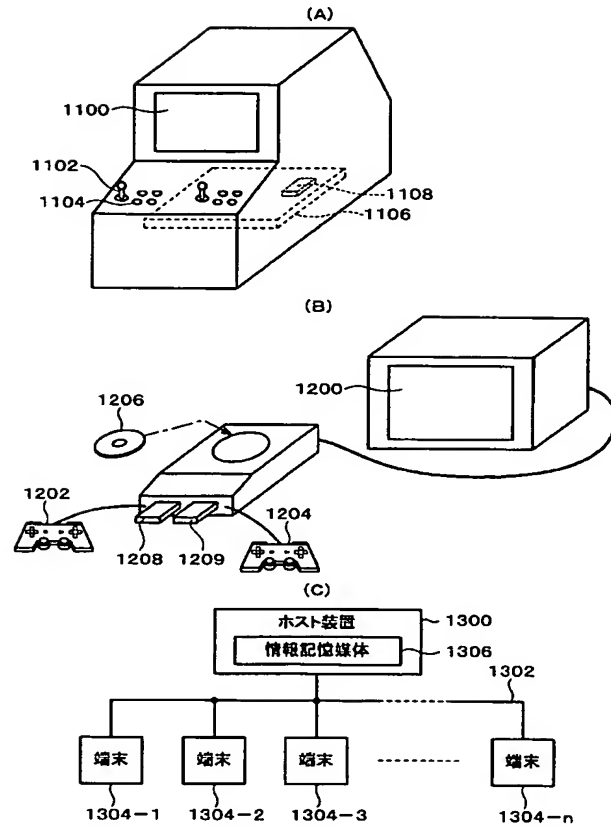
【図17】



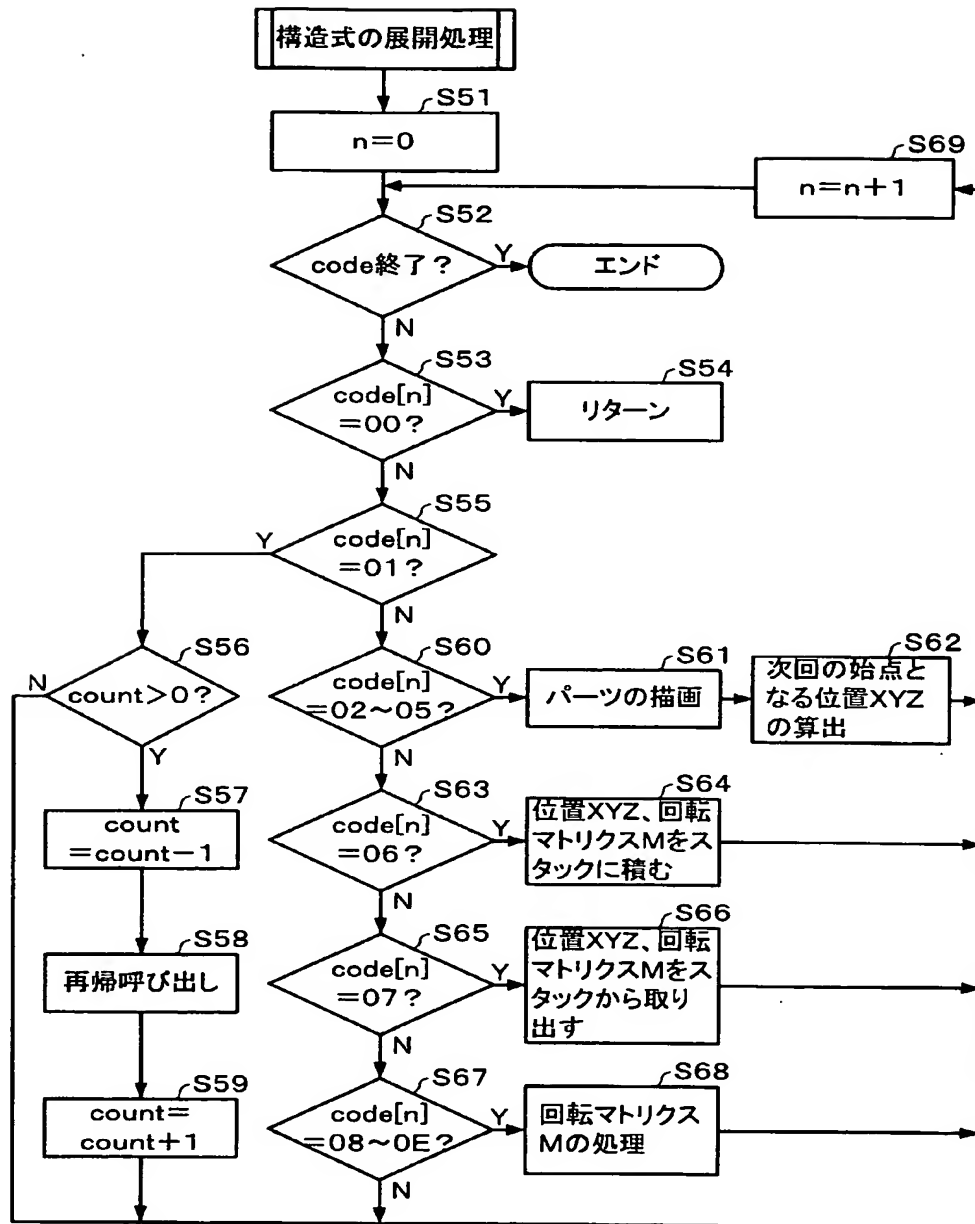
【図18】



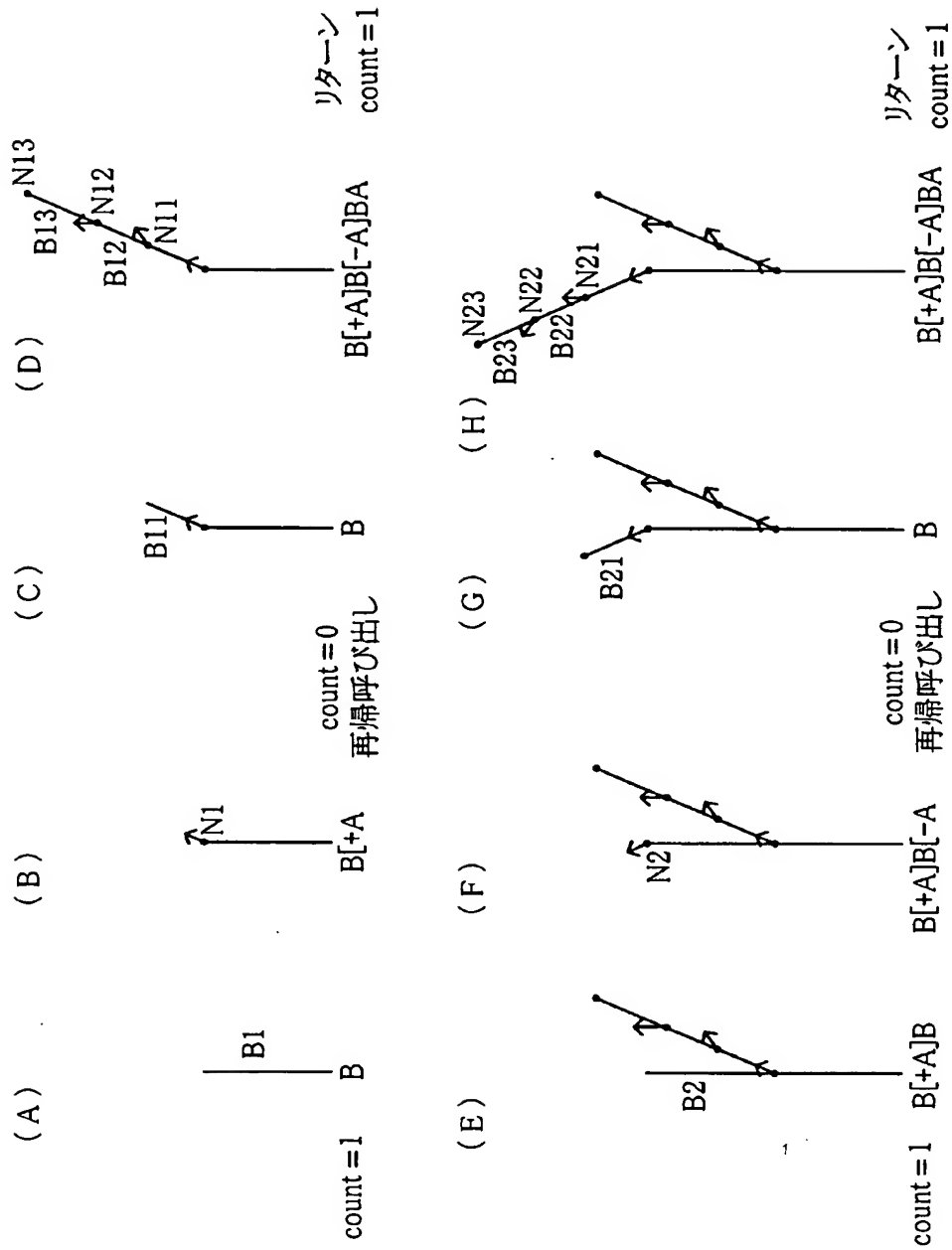
【図23】



【図19】



【図20】



【図22】

